

DAM  
Desarrollo de Aplicaciones Multiplataforma  
2º Curso

AD  
Acceso a Datos

UD 3  
Manejo de Conectores  
(JDBC)

IES BALMIS  
Dpto Informática  
Curso 2019-2020  
Versión 1 (09/2019)

## 2. EJERCICIOS Java y MySQL

### UD3Ejer201 - BDTAREAS

Realiza un programa en Java que acceda a MySQL y muestre los datos introducidos en la tabla **tareas** de la BD **bdtareas**.

Los campos a mostrar son **codigo**, **descripcion**, **fecha\_prevista** y **terminada**.

Para imprimir pretty puedes usar **printf**

<https://www.baeldung.com/java-printstream-printf>

La salida prevista es:

Código	Descripción	Fecha prevista	Terminada
1	Repasar SQL	10/10/2019	NO
2	Instalar NetBeans	11/10/2019	NO
3	Realizar copia de seguridad	10/10/2019	NO
4	Examinar en busca de virus	03/10/2019	SI

### UD3Ejer202 - BDAPPS

Crea desde Java una nueva tabla denominada **ejecuciones** dentro de la base de datos **bdapps**, que permita guardar datos sobre un programa con los siguiente campos:

**id (autonumerico)**, **codprograma**, **fecha** y **usuario**

Captura la excepción **SQLException** por si lo ejecutamos dos veces:

```
try {
    ...
} catch (SQLException e) {
    System.out.println(e.getMessage());
}
```

### UD3Ejer203 - BDAPPS

Desde Java, añade a la tabla de **programas** un nuevo campo, que indicará su tamaño medido en KB denominado **ocupacion** de tipo entero, con valor por defecto 0.

### UD3Ejer204 - BDAPPS

Crea desde Java un programa que

- actualice el valor de **ocupacion** para cada programa y
- añada filas en la tabla de **ejecuciones**. Al menos uno por cada programa, y de alguno de ellos dos.

Comprueba el valor devuelto por el método, para asegurarte de que todas se han insertado correctamente.

**UD3Ejer205 - BDAPPS**

Crea una función en PL/SQL denominada **nomprograma**, que dado el código de un programa devuelva su nombre. Escribe este código en **nomprograma.sql** y ejecuta este código en Workbench para que la función esté disponible.

Crea un programa en Java que pida al usuario el código de un programa y muestre su nombre, usando la función anterior.

```
/* Eliminar la función si ya existe */
DROP FUNCTION IF EXISTS nomprograma;

/* Crear la función */
DELIMITER //
CREATE FUNCTION nomprograma(pCodigo INT)
    RETURNS VARCHAR(60)
BEGIN
    DECLARE resul VARCHAR(60);

    SELECT nombre INTO resul FROM programas WHERE codigo=pCodigo;

    RETURN resul;
END //
DELIMITER ;

/* Prueba desde Workbench */
SELECT nomprograma(1);
```

**UD3Ejer206 - BDAPPS**

Crea una función en PL/SQL denominada **numejecuciones**, que dada una fecha, devuelva la cantidad de ejecuciones realizadas. Escribe este código en **numejecuciones.sql** y ejecuta este código en Workbench para que la función esté disponible.

Crea un programa en Java que pida una fecha y muestre la cantidad de ejecuciones de ese día.

```
/* Eliminar la función si ya existe */
DROP FUNCTION IF EXISTS numejecuciones;

/* Crear la función */
DELIMITER //
CREATE FUNCTION numejecuciones(pFecha DATE)
    RETURNS INT
BEGIN
    DECLARE resul INT;

    SELECT COUNT(*) INTO resul FROM ejecuciones WHERE fecha=pFecha;

    RETURN resul;
END //
DELIMITER ;

/* Prueba desde Workbench */
SELECT numejecuciones('2019-10-03');
```

**UD3Ejer207 - BDAPPS**

Crea un programa en Java que recorra todas los programas y los almacene en un `ArrayList<Programa>`.

Deberá mostrarse en pantalla los datos de los programas que se encuentren en la unidad "C:". Se utilizará el método `toString` de la clase `Programa` y el formato será JSON.

```
{"codigo":1, "nombre":"LibreOffice", "carpeta":"C:/Program Files/LibreOffice/program" }  
{"codigo":2, "nombre":"Mozilla Thunderbird", "carpeta":"C:/Program Files (x86)/Mozilla  
Thunderbird" }
```

**UD3Ejer208 - BDAPPS**

Crea un programa que pida al usuario datos de **programas** y los añada a la base de datos.

*En caso de que se introduzca un código repetido, el programa debe comportarse de forma amigable: no interrumpir la ejecución, sino pedir un nuevo código y guardar los datos (supondremos que el usuario sabe qué datos va a introducir pero se está "inventando" los códigos, de modo que el problema no se tratará de que ha tratado de volver a introducir una ficha que ya estaba, sino que es una ficha nueva, en la que por error ha reutilizado un código que ya había empleado, pero el resto de datos sí son correctos).*